Bank Note Authentication

Dipanshu Sharma
Department of Mathematics
Indian Institute of Technology, Bombay
18B090004@iitb.ac.in

Ayushmaan Dev Verma
Department of Mathematics
Indian Institute of Technology, Bombay
18B090002@iitb.ac.in

Abstract

Banknote forgery is an issue faced by many banking institutions around the globe. This paper reviews our work to build a technique based on machine learning capable of detecting such forgery with high accuracy. We went through various stages for building this technique, starting from downloading the banknote specifications data to building a highly accurate model. Next, we explored various front-end deployment methods so that the model can be used on a larger scale. Finally, we created a user-friendly web-based front-end for our model to accept user input in various ways.

Keywords—banknote forgery, machine learning, model, front-end

Introduction

Forged notes, also called counterfeit currency, is produced without the legal sanction of the State or government in an attempt to imitate that currency and deceive the recipient [1]. Counterfeit currency has several adverse effects on society. These are— reduction in the value of the money, increase in prices due to inflation, decrease in trust and confidence over the banknotes, and losses for those who inadvertently accepted fake notes.

Currently, the anti-counterfeiting measures are strict control over specific security features (like holograms and fluorescence) of the notes and printing raised intaglio (images printed with ink) on the notes. Using both, experts can detect if a forged note has entered the system. However, the issue with this is that detecting forgery by hand is a very lengthy and error-prone process, and for the past few years, fake currency in circulation has been increasing [2].

Supervised machine learning techniques, namely, Decision Trees, Random Forest, Gradient Boosting and Logistic Regression Classifiers, were implemented. The dataset from the UCI Machine Learning Repository was collected by extracting essential features using the wavelet transform tool to train these models. The class (forged or authentic) of each sample was also imputed in the collected data. A comparative study of these techniques concerning their accuracy is also shown. Finally, Pickle, Flask and Flasgger were used for the front-end development, and the end-to-end tool was deployed on Postman. [3]

Background and Previous Work

One should have a grasp of an intermediate level of Python to analyse the source code. Additionally, for understanding the front-end, one should be comfortable with libraries such as Flask, Pickle and Flasgger. Also, one should know about Postman to visualise the results of the front-end process. Beyond Basic ML, one should be well-versed in hyperparameter tuning of different machine learning models.

In the resources we analysed, everyone generally focussed upon only the machine learning and model building part of the project, and no one had worked upon the

front-end development and deployment of their models [4]. Also, the project involves hyperparameter tuning, which was not used for model selection in the earlier works. Hyperparameter tuning is the process of finding a set of optimal hyperparameters for a machine learning algorithm. The same model can require different constraints, weights or learning rates to generalise different data patterns. These values are called hyperparameters and they must be tuned so that the model can optimally solve the assigned machine learning problem. Hyperparameter tuning finds the set of hyperparameters that yields an optimal model which maximises the accuracy on the given dataset. We used cross-validation to estimate the generalised performance. [5]

Datasets

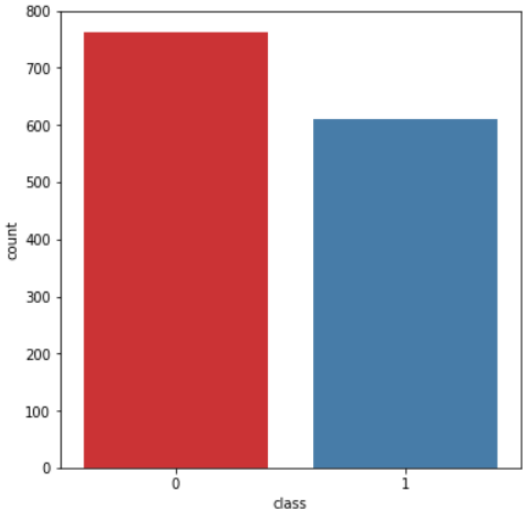
Data Information: The dataset was extracted from images that were taken from authentic and forged banknotes. For digitisation, an industrial-grade camera was used. The final images are of size 400 x 400 pixels. Due to the object lens and distance to the studied object, grey-scale pictures with a resolution of about 660 dpi were gained. Finally, the Wavelet Transform tool was used to extract features from images.

Feature Information: Variance of Wavelet Transformed Image (continuous); Skewness of Wavelet Transformed Image (continuous); Kurtosis of Wavelet Transformed Image (continuous); Entropy of Image (continuous); and Class (integer).

Source: UCI Machine Learning Repository; Volker Lohweg (University of Applied Sciences, Ostwestfalen-Lippe, volker.lohweg '@' hs-owl.de) [6]

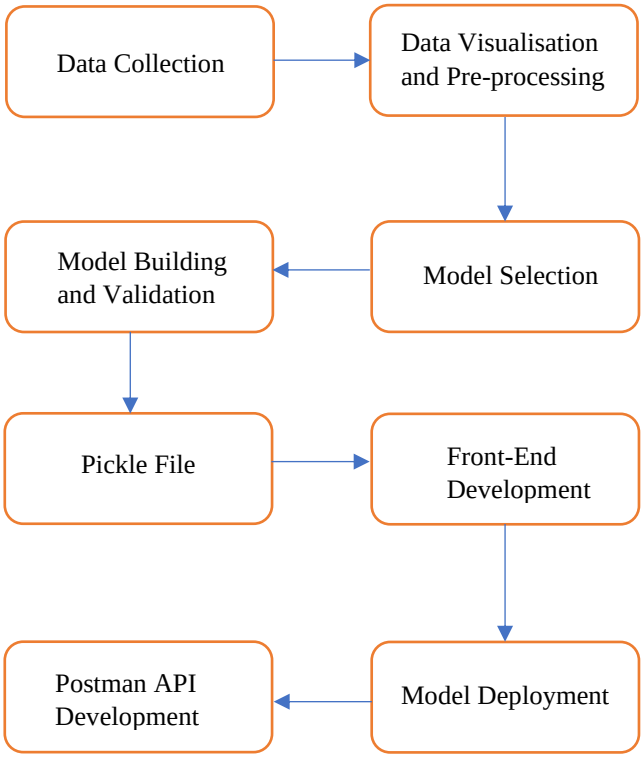
Challenges: No challenges were faced while procuring the data because it is very well documented and has been pre-cleaned to have no null/missing values.

Fig. 1: Our dataset had a fairly equitable distribution, with 762 (class = 0) forged and 610 (class = 1) authentic notes.



Procedure and Experiments

Fig. 2: Flowchart of the procedure



First, we collected the data from the UCI Machine Learning Repository. After verifying that the data is clean, we visualised and pre-processed the dataset using count-plots, pair-plots, distribution plots, scatter matrices, heatmaps and box-plots. As both classes have an almost equal number of samples, the model was robust for both kinds of samples. Next, we selected four classifiers for the model selection and analysis, namely, Decision Trees, Random Forests, Gradient Boosting and Logistic Regression. For each classifier, we performed hyperparameter tuning for at least one hyperparameter. Based on the accuracy of the selected models, we chose the model with the best accuracy. Now, we stored the chosen model in a *pickle* file using serialisation for further front-end development. For the front-end, we use Flask and Flasgger libraries of Python, in which we gave two options (routes) for the user to input the values manually. Either they can input individual values for a feature or upload a CSV file containing the values, after which the user can predict whether the sample is forged or not. We also visualised the front-end based on Postman API Development.

Explicitly, we used two experiments to test the utility of our technique. First, for the model selection, we used accuracy (cross-validation score) [7] to choose the best performing model. Secondly, we imputed sample data into our front end to confirm whether our front-end API was giving us the correct output or not.

Results

After both hyperparameter tuning and cross-validation, the best possible accuracy was given by the Gradient Boosting Algorithm with the number of estimators to be 150. Our accuracy came to be more than 99.7% which was better than other models perused.

Secondly, our API was working perfectly well, delivering the same output as our selected model. This is also achieved faster because of no need to create arrays or data frames of input data by the user to run the predict function and get the output.

Classifier	**Average Accuracy (approximate)**	**Maximum Accuracy (approximate)**
Decision Trees	0.978	0.982
Random Forest	0.993	0.994
Gradient Boosting	0.994	0.997
Logistic Regression	0.989	0.990

Conclusions

Banknote forgery detection is an essential need. It is not easy to manually detect forged banknotes, and machine learning models can help in this regard. In this report, we have explained how we solved the problem of banknote forgery detection using machine learning models. We compared four different classifiers in terms of accuracy (cross-validation score) and concluded that the Gradient Boosting classifier (with 150 estimators) was the most suitable algorithm with more than 99.7% accuracy. This technique is an efficient way of solving the problem for all banking machines that accept all types of notes. In the future, we can improve the performance by collecting more and more data and improving our image capturing methods. Also, we can make it more user-friendly for the front-end portion by hosting a separate mobile or web application. We can also use sophisticated deployment methods like Docker so that the application can be hosted on other systems. Moreover, we can make more options for inputting data, such as excel or text files.

Statement of Contributions

Dipanshu Sharma was responsible for the project's conception, data collection, cleaning, pre-processing, and creating the pickle file. Ayushmaan Dev Verma carried out data visualisation, model selection, hyperparameter tuning and model testing. Also, Dipanshu Sharma was responsible for the front-end development using Flask and Flasgger. Ayushmaan was responsible for the Postman API Development and the project report-writing as well. Both worked on the video-making and editing together.

References

[1]https://en.wikipedia.org/wiki/Counterfeit_money
[2]https://www.thehindu.com/news/national/most-fakes-are-2000-notes-ncrb/article32743045.ece
[3a]https://en.wikipedia.org/wiki/Supervised_learning
[3b]https://www.geeksforgeeks.org/supervised-unsupervised-learning/
[3c]http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[3d]http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[3e]http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[3f]http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[3g]https://docs.python.org/3/library/pickle.html
[3h]https://en.wikipedia.org/wiki/Flask_(web_framework)
[3i]https://flask.palletsprojects.com/en/2.0.x/
[3j]https://pypi.org/project/flasgger/0.5.4/
[3k]https://medium.com/analytics-vidhya/flasgger-an-api-playground-with-flask-and-swagger-ui-6b6806cf8884
[3l]https://stackoverflow.com/questions/41604212/how-to-use-flasgger-with-flask-applications-using-blueprints
[3m]https://www.postman.com/api-network/
[3n]https://www.postman.com/explore/?utm_source=postman-website&utm_medium=referral
[4a]https://www.researchgate.net/publication/279205560_BANKNOTE_AUTHENTICATION_USING_ARTIFICIAL_NEURAL_NETWORK
[4b]https://machinelearningmastery.com/neural-network-for-banknote-authentication/
[4c]https://medium.com/geekculture/building-a-machine-learning-model-for-banknote-authentication-f9c6855a9057
[4d]https://www.neuraldesigner.com/learning/examples/banknote-authentication
[5]https://en.wikipedia.org/wiki/Hyperparameter_optimization
[6]https://archive.ics.uci.edu/ml/datasets/banknote+authentication
[7]http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html